

PROJET SYSTEME D'EXPLOITATION

Simulation de la catastrophe de Tchernobyl en un programme C

Housseem Tlili, Hassen Tahri, et Ali Hosni Bouslimi

Rapport de 4 pages

INTRODUCTION

CONCEPTION ET PRINCIPE DE RESOLUTION

- CONCEPTIONS
- PRINCIPE DE RESOLUTION

REALISATION

- ENVIRONNEMENT MATERIEL
- DIFFICULTES RENCONTRES
- PRESENTATION

CONCLUSION

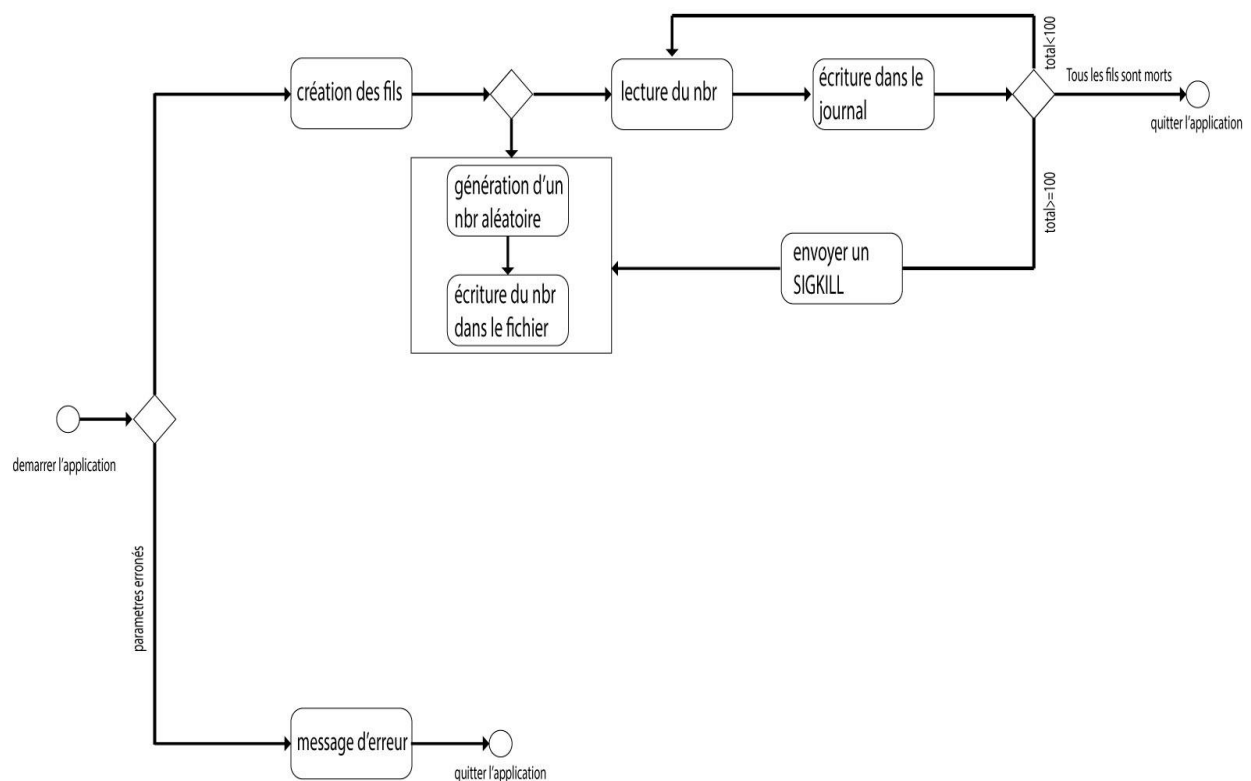
INTRODUCTION

Catastrophe de Tchernobyl L'accident nucléaire de l'usine de Tchernobyl s'est produite le 26 avril 1986. C'était la plus grande catastrophe de l'énergie nucléaire de l'histoire. L'explosion a eu lieu dans le quatrième bloc de la centrale de Tchernobyl, située à seulement 120 km de la capitale d'Ukraine - Kiev, près de la frontière avec la Biélorussie.

Dans ce projet, on veut simuler l'explosion de la centrale nucléaire de Tchernobyl en utilisant les processus et ces fonctionnalités dans les systèmes UNIX.

CONCEPTION ET PRINCIPE DE RESOLUTION

CONCEPTIONS :



PRINCIPE DE RESOLUTION

A partir des données initiales, notre programme se charge de créer les n capteurs de danger. Ces derniers vont communiquer avec le programme "père" via un fichier commun.

Ce programme accepte en entrée le nom du fichier 'journal' et ne nombre de capteurs à crée.

La première étape consiste à créer les capteurs de danger, remplir un tableau du structure capteur contenant les pids de chacun des capteurs créés. Le programme 'père' se met en veille en attendant que l'un des capteur écrit dans le fichier commun de communication.

La deuxième étape commence dès que le premier capteur écrit un nombre aléatoire dans le fichier. Le père se réveille donc et commence à lire les niveaux de danger générés par les capteurs et met à jour le nombre totale des niveaux de danger générés par chaque capteur. Si le nombre totale d'un capteur dépasse 100 le père l'envoie un signale pour le terminer et le note dans son journal.

Une fois que le nombre des capteurs terminés est égal aux nombre des capteurs créés initialement le programme 'père' calcule le dépassement total et son moyenne.

REALISATION

ENVIRONNEMENT MATERIEL :

Pour la réalisation de notre application, un ordinateur portable a été utilisé. Il a les caractéristiques suivantes :

Marque	DELL
Processeur	Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz 2.50 GHz
Mémoire installée	4,00 GB
Système d'exploitation	Xubuntu

ENVIRONNEMENT LOGICIEL :

Nous présenterons brièvement les logiciels que nous avons employés au cours de notre projet.

Geany



Geany est un EID gratuit de C et d'autres langages fait pour satisfaire les besoins les plus courantes des développeurs. Il est construit pour être très extensibles et totalement configurable.

Mousepad



Mousepad est un Gnome éditeur de texte. Au temps qu'il est très simple et facile à utiliser, mousepad est un éditeur de texte puissant pour des propos générales.

DIFFICULTES RENCONTRES :

1-Problème: Bloquer le processus père jusqu'à ce que le premier fils écrive dans le fichier 'niveau_dangers.txt'.

Résolution : Utilisation de la méthode pause() dans le père pour mettre le programme père 'en veille' jusqu'à l'arrivée d'un signal. Ainsi chaque processus fils envoie un signal SIGUSR1 à son père après avoir écrit dans le fichier

2-Problème : Comment génère des nombres aléatoires différents à chaque fois pour chacun des processus fils.

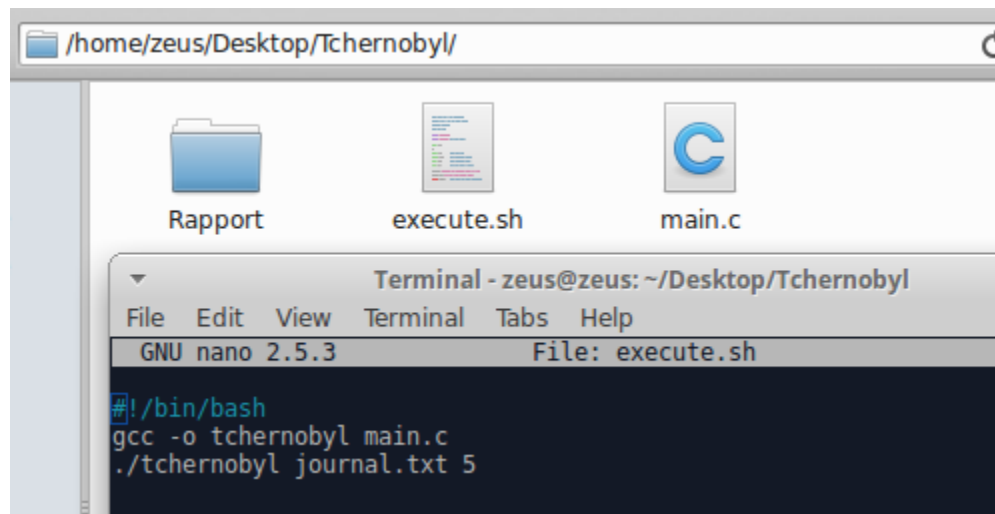
Résolution : Utilisation du pid (par appel à la fonction getpid()) comme 'graine' pour la génération d'une nouvelle séquence de nombres aléatoires. Ces séquences sont reproductibles en appelant **srand()** avec une valeur différente de graine (le pid)

GESTION DES TACHES :

Tache	Horaire	Etudiant
Codage	6 heures	Houssem, Hassen, Ali
Conception	1 heure	Houssem
Tests	1 heure	Hassen
Rédaction de rapport	2 heures	Ali

PRESENTATION :

Compilation et exécution du programme.



Exemple du fichier 'niveau_danger.txt' dont les fils sont les écrivains et le père est le lecteur.

```
File Edit Search View Document Help
3858 2
3857 4
3859 2
3860 4
3856 9
3859 3
3857 3
3860 3
3856 3
3858 4
3857 5
3858 1
```

Exemple du fichier 'journal.txt' passé en argument au programme et dont le père est l'écrivain.

```
File Edit Search View Document Help
fils: 3856 niveau: 4 total: 99
fils: 3858 niveau: 8 total: 107
Un signal SIGKILL est envoyey au fils 3858
fils: 3860 niveau: 2 total: 100
Un signal SIGKILL est envoyey au fils 3860
fils: 3857 niveau: 4 total: 90
fils: 3859 niveau: 8 total: 107
Un signal SIGKILL est envoyey au fils 3859
fils: 3856 niveau: 8 total: 107
Un signal SIGKILL est envoyey au fils 3856
fils: 3857 niveau: 1 total: 91
fils: 3857 niveau: 2 total: 93
fils: 3857 niveau: 2 total: 95
fils: 3857 niveau: 2 total: 97
fils: 3857 niveau: 1 total: 98
fils: 3857 niveau: 4 total: 102
Un signal SIGKILL est envoyey au fils 3857
Le total des depassement est 23
```

CONCLUSION

Dans le cadre de ce mini-projet, nous avons pu développer un programme de simulation de l'explosion de la centrale nucléaire du Tchernobyl. Cette application nous permettra d'appliquer nos connaissances sur la gestion des processus dans les système UNIX tout en exploitant la communication interprocessus via les signaux.

Ce projet était une occasion pour exploiter les autres outils de communication interprocessus IPC comme les files de messages, les verrous, les sémaphores et les mutex.